Assignment No: 02

Name of student:

Roll No:

Practical Batch:

Title of the Assignment: A* algorithm for 8 puzzle problem

Problem statement: Implement A star Algorithm for any game search problem.

Objective:

- To understand the concept of Informed search techniques.
- To implement A* algorithm for 8 puzzle game problem.

Theory:

A* algorithm:

A* is a computer algorithm that is widely used in path finding and graph traversal, the process of plotting an efficiently traversable path between multiple points, called nodes.
Noted for its performance and accuracy, it enjoys widespread use.
The key feature of the A* algorithm is that it keeps a track of each visited node which helps in ignoring the nodes that are already visited, saving a huge amount of time. It also has a list that holds all the nodes that are left to be explored and it chooses the most optimal node from this list, thus saving time not exploring unnecessary or less optimal nodes.
Here we use two lists namely 'open list 'and 'closed list'.The open list contains all the nodes that are being generated and are not existing in the closed list and each node explored after it's neighboring nodes are discovered is put in the closed list and the neighbors are put in the open list this is how the nodes expand. Each node has a pointer to its parent so that at any given point it can retrace the path to the parent. Initially, the open list holds the start (Initial) node. The next node chosen from the open list is based on its **f score**; the node with the least f score is picked up and explored.

f-score = h-score + g-score

A* uses a combination of heuristic value (h-score: how far the goal node is) as well as the g-score (i.e. the number of nodes traversed from the start node to current node).

- A* Algorithm is one of the best path finding algorithms.
- But it does not produce the shortest path always.
- This is because it heavily depends on heuristics.

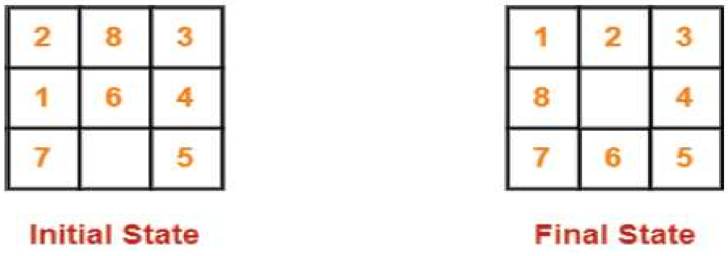
Algorithm:

1. Create a search graph G, consisting solely of the start node, no. Put no on a list called OPEN.
2. Create a list called CLOSED that is initially empty.
3. If OPEN is empty, exit with failure.
4. Select the first node on OPEN, remove it from OPEN, and put it on CLOSED. Called this node n.
5. If n is a goal node, exit successfully with the solution obtained by tracing a path along the pointers from n to no in G. (The pointers define a search tree and are established in Step 7)
6. Expand node n, generating the set M, of its successors that are not already ancestors of n in G. Install these members of M as successors of n in G.
7. Establish a pointer to n from each of those members of M that were not already in G (i.e., not already on either OPEN or CLOSED). Add these members of M to OPEN. For each member, m, of M that was already on OPEN or CLOSED, redirect its pointer to n if the best path to m found so far is through n. For each member of M already on CLOSED, redirect the pointers of each of its descendants in G so that they point backward along the best paths found so far to these descendants.
8. Reorder the list OPEN in order of increasing f values. (Ties among minimal f values are resolved in favor of the deepest node in the search tree.)
9. Go to Step 3.

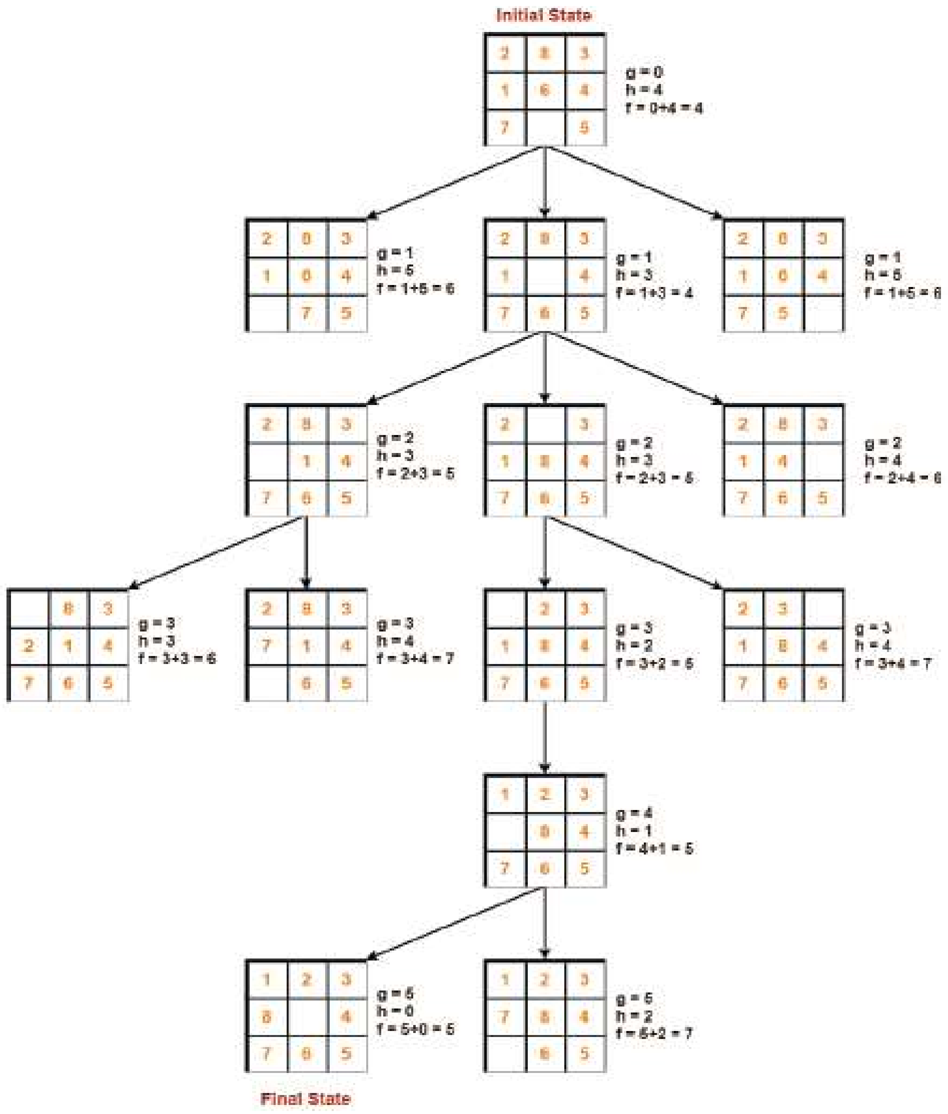
8-Puzzle problem:

In our 8-Puzzle problem, we can define the **h-score** as the number of misplaced tiles by comparing the current state and the goal state or summation of the Manhattan distance between misplaced nodes.
g-score will remain as the number of nodes traversed from a start node to get to the current node. We can calculate the **h-score** by comparing the initial (current) state and goal state and counting the number of misplaced tiles.

Given an initial state of a 8-puzzle problem and final state to be reached-



Find the most cost-effective path to reach the final state from initial state using A* Algorithm. Consider g(n) = Depth of node and h(n) = Number of misplaced tiles.



Conclusion:

We have implemented BFS and DFS using recursive algorithm for undirected graph.

Oral questions:

1. What are informed and uninformed search techniques?
2. List differences between informed and uninformed search techniques?
3. What is the evaluation function of A* algorithm?
4. What are the applications of A* algorithm?
5. What is problem solving agent?
6. Is A* algorithm informed or uninformed search technique?
7. What is Best first search algorithm?
8. Where are searching techniques applied in AI?
9. What are Blind search algorithms?
10. What is meant by Heuristic?